

20. (original) A method as recited in claim 17, wherein when it is determined that the type of token is a process loop type token, the loop process having a loop entry condition and a loop exit condition suitable for entering and exiting, respectively, a corresponding loop expression and a loop body, the converting the process further comprises:

- allocating a query CN to the process block CN list;
- recursively converting a loop expression sub-parse tree to a corresponding loop expression ADD; and
- recursively converting a loop body sub-parse tree to a loop body ADD.

21. (original) A method as recited in claim 20, wherein when it is determined that the process token identifies a process loop, the annotating the ADD further comprises:

- annotating the loop expression ADD with appropriate loop expression control nodes;
- mapping the query CN to the loop expression ADD;
- annotating the loop body ADD with appropriate loop body control nodes;
- returning a loop body first CN and a loop body last CN;
- allocating a null CN;
- pointing the loop body last CN to the null CN;
- pointing the loop entry condition to the loop body first CN;
- pointing the loop exit condition to the null CN; and
- identifying the query CN as a first CN and the null CN as a last CN.

22. (original) A method as recited in claim 17, wherein when it is determined that the type of token is a process suspend type token, the converting the process further comprises:

- recursively converting a suspend sub-parse tree to a corresponding suspend ADD;

put a period here

Please
enter
this
amendment
DMC